REMARKS/ARGUMENTS

Upon entry of this amendment, claims 1-3, 8, 10-12, 17, 19, 21, 26, and 28 will be amended, and claims 1-32 will remain pending. Of these, claims 1-8 and 10-17 stand rejected under 35 U.S.C. § 103 as being unpatentable over Hao (U.S. Patent No. 4,569,016). Claims 9 and 18 stand under 35 U.S.C. § 103 as being unpatentable over Hao in view of Kabir (U.S. Patent No. 5,933,160). Claims 19-23 and 26-30 stand rejected under 35 U.S.C. § 103 as being unpatentable over Hao. Claims 24, 25, 31, and 32 stand rejected under 35 U.S.C. § 103 as being unpatentable over Hao in view of Kabir.

Rejections Under 35 U.S.C. § 103

Patentability of Claims 1-32

Claim 1

Applicants have amended claim 1. As amended, claim 1 is patentable over the cited reference Hao. This is because Hao fails to teach or suggest the claimed feature of a mask comprising a plurality of mask fields that are each independently selectable as either a write-enabled mask field or a mask-disabled mask field. Claim 1 recites:

> **"the mask comprising a plurality of mask fields that each corresponds to a data field of the data contained in the at least one register, each of the plurality of mask fields being independently selectable as either a write-enabled mask field or a write-disabled mask field..." (emphasis added).**

This claim language requires that each of the plurality of mask fields must be independently selectable as either a write-enabled mask field or a write-disabled mask field. For example, if one mask field is selected as a write-enabled mask field (e.g., indicated as a logic "1"), an adjacent mask field cannot be dictated to be a write-disabled mask field (e.g., indicated as a logic "0"). Instead, the adjacent mask field must be independently selectable as either a write enabled mask field or a write-disabled mask field. That is, the value of one mask field cannot dictate the value of another mask field – i.e., they must be independently selectable.

Hao teaches the exact opposite of this claimed feature. Hao discloses two types of masks: (1) a three-part mask ("a substring of ones surrounded by zeros" or "a substring of zeros surrounded by ones") and (2) a two-part mask ("ones followed by zeros" or "zeros followed by ones"). See Hao at col. 26, lines 3-46, as cited by the Office Action dated 10/30/07, pp. 2-3, paragraph 3. Both types of masks require that a substring be defined within the mask, and that the rest of the mask be opposite in value to the substring. For example, a three-part mask that is "a substring of ones surrounded by zeros" might appear as:

"00000000111111111111111111110000" (32-bit example)

This mask has three parts: (a) a first part comprising 8 bits of zeros, (b) a second part comprising 20 bits of ones, and (c) a third part comprising 4 bits of zeros. Here, once the value of the second part (i.e., the substring) of the mask is selected, the values of the remaining two parts are immediately dictated because they must be opposite in value. Thus, if the second part is selected as ones (20 bits of ones), the first part cannot be independently selected – it must be all zeros (8 bits of zeros), and the third part cannot be independently selected – it must also be all zeros (4 bits of zeros).[1] The same is true for Hao's two-part mask. For example, a two-part mask that is "1's followed by 0's" might appear as:

"11111111111111111111000000000000" (32-bit example)

This mask has two parts: (a) a first part comprising 20 bits of ones and (b) a second part comprising 12 bits of zeros. Here, once the value of the first part (i.e., the substring) is selected, the value of the second part is immediately dictated because it must be opposite in value. Thus, if the first part is selected as ones (20 bits of ones), the second part cannot be independently selected – it must be all zeros (12 bits of zeros).[2]

[1] Hao also discloses that the second part can be selected as zeros ("a substring of zeros surrounded by ones"). In that case, the parts of the mask are again not independently selectable. For example, if the second part is selected as zeros (20 bits of zeros), the first part cannot be independently selected – it must be all ones (8 bits of ones), and the third part cannot be independently selected – it must also be all ones (4 bits of ones).

[2] Hao also discloses that the first part can be selected as zeros ("0's followed by 1's"). In that case, the parts of the mask are again not independently selectable. For example, if the first part is selected as zeros (20 bits of zeros), the second part cannot be independently selected – it must be all ones (12 bits of ones).

As these examples amply illustrate, both types of masks taught by Hao ensure that once the value of one portion of the mask is selected, the value of the rest of the mask cannot be independently selected, because it would be dictated to take on the opposite value as the selected portion. The portions defined in Hao's masks, be it a three-part mask or a two-part mask, are certainly not "independently selectable as either a write-enabled mask field or a mask-disabled mask field," as recited in claim 1. For at least this reason, claim 1 is patentable over Hao.

Claim 10

As amended, claim 10 is distinguished over Hao for similar reasons as discussed above with respect to claim 1. Claim 10 recites a data processing system having an execution unit capable of operation using a mask comprising a plurality of mask fields that are each independently selectable as either a write-enabled mask field or a mask-disabled mask field. As explained with respect to claim 1, Hao fails to disclose and in fact teaches away from such a claimed feature. Thus, claim 10 is also clearly patentable over Hao.

Claim 19

As amended, claim 19 is patentable over Hao. Claim 19 recites:

> **"the execution unit capable of performing a bitwise insert operation that operates on a first and a second operand stored in at least one register in the register file, wherein each bit in the second operand is individually selectable as either having a first predetermined value or a second predetermined value, wherein for each bit in the first operand, the bitwise insert operation inserts the bit into a corresponding bit position in a destination value if a corresponding bit in the second operand has the first predetermined value..." (emphasis added).**

This requires a bit insert operation that inserts bits from a first operand (into a destination value) based on corresponding bits in a second operand, such that each bit in the second operand must be independently selectable as either having a first predetermined value or a second predetermined value. For example, if one bit in the second operand is selected as the first predetermined value (e.g., indicated as a logic "1"), an adjacent bit in the second operand cannot be dictated to be the second predetermined value (e.g., indicated as a logic "0"). Instead, the adjacent bit must be independently selectable as either the first predetermined value or the second predetermined value. That is, the value of one bit in the second operand cannot dictate the value of another bit in the second operand – i.e., they must be independently selectable.

Again, Hao teaches the exact opposite of this claimed feature. As discussed above with respect to claim 1, Hao teaches two types of mask: (1) a three-part mask ("a substring of ones surrounded by zeros" or "a substring of zeros surrounded by ones") and (2) a two-part mask ("ones followed by zeros" or "zeros followed by ones"). Both types of masks require that a substring be defined within the mask, and that the rest of the mask be opposite in value to the substring. For the three-part mask, if the second part is selected to as ones, the first part cannot be independently selected – it must be all zeros, and the third part cannot be independently selected – it must also be all zeros. For the two-part mask, if the first part is selected as ones, the second part cannot be independently selected – it must be all zeros. Thus, both types of masks taught by Hao ensure that once the value of one portion of the mask is selected, the value of the rest of the mask cannot be independently selected, because they would be dictated to take on the opposite of the value of the selected portion. The bits in Hao's masks, be it a three-part mask or a two-part mask, are certainly not "independently selectable."

Accordingly Hao not only fails to disclose, but in fact teaches away from, a bit insert operation that inserts bits from a first operand (into a destination value) based on corresponding bits from a second operand, such that each bit in the second operand must be independently selectable as either having a first predetermined value or a second predetermined value, as recited in claim 19. For at least this reason, claim 19 is patentable over Hao.

Claim 26

As amended, claim 26 is clearly patentable over Hao for similar reasons as discussed above with respect to claim 19. Claim 26 recites a device having a programmable processor comprising an execution unit capable of performing a bitwise insertion operation using a first and a second operand, wherein insertion of bits from the first operand is based on bits in the second operand. Specifically, each bit in the second operand is individually selectable as either having a first predetermined value or a second predetermined value. As explained with respect to claim 19, Hao fails to disclose and in fact teaches away from such a claimed feature. Thus, claim 26 is also patentable over Hao.

Dependent claims 2-9, 11-18, 20-25, and 27-32

Dependent claims 2-9, 11-18, 20-25, and 27-32 depend from independent claims 1, 10, 19, and 26, respectively, and therefore incorporate the limitations of their respective independent claims. As discussed above, the primary reference Hao fails to disclose and in fact teaches away from certain of these limitations. The secondary reference Kabir does not make up for such deficiencies. Thus, for at least the same reasons stated above regarding their respective independent claims, dependent claims 2-9, 11-18, 20-25, and 27-32 are also patentable over the cited references.

Additional Reasons for Patentability of Claims 9, 18, 24, 25, 31, and 32:
Failure of Kabir to Qualify as Prior Art

Furthermore, claims 9, 18, 24, 25, 31, and 32 are also patentable on a separate and additional ground – Kabir cannot qualify as prior art against the present invention. Claims 9, 18, 24, 25, 31, and 32 stand rejected based on combination of the primary reference Hao and the secondary reference Kabir. However, Kabir fails to qualify as "prior art" with respect to the present invention. The present application claims priority back to the 8/16/95 filing date of U.S. Patent Application No. 08/516,036, which issued as U.S. Patent No. 5,742,840 (the '840 patent). This chain of priority also includes U.S. Patent Application No. 09/382,402, which issued as U.S. Patent No. 6,295,599 (the '599 patent). By contrast, the earliest possible filing date for Kabir is 11/27/95. Thus, Kabir cannot qualify as prior art because its earliest possible filing date is later than the priority date of the present application.[3]

It appears that the Patent Office is satisfied with the detailed explanation that the Applicants provided in the previous response regarding how the '840 disclosure and '599 disclosure provide support for the claims, at least with regard to masking and bitwise insertion

[3] Further details on the priority chain of both the present invention and the Kabir reference are provided in the Applicants' prior responses filed 8/16/07 and 11/15/06.

features.[4] However, with respect to the additional limitations recited in claims 9, 18, 24, 25, 31, and 32, the Office Action contends that support has yet to be pointed out in the '840 and '599 disclosures.[5] To address the Examiner's concerns, Applicants have identified proper support found in the '840 and '599 disclosures for the additional limitations in claims 9, 18, 24, 25, 31, and 32. This is explained in detail below and further evidenced by a declaration from Mr. Korbin Van Dyke, submitted as part of this Response.

Claims 9 and 18 each recites limitations relating to "in response to decoding a second single instruction specifying a register containing a first plurality of floating-point operands and another register containing a second plurality of floating-point operands, multiply the first plurality of floating-point operands by the second plurality of floating-point operands to produce a plurality of products and provide the plurality of products to partitioned fields of a result register as a catenated result." The '840 and '599 disclosures clearly describe embodiments of such an instruction. The '840 and '599 disclosures each includes an appendix filed as part of the original application. Both the '840 appendix and the '599 appendix describe embodiments of a "single instruction" that performs multiplication of a first plurality of floating-point operands by a second plurality of floating-point operands to produce a plurality of products provided as a catenated result, as recited in claims 9 and 18. In fact, at least three such instructions are described: (1) "GF.MUL.16" (Group floating-point multiply half), (2) "GF.MUL.32" (Group floating-point multiply single), and (3) "GF.MUL.64" (Group floating-point multiply double). These instructions are identified at pp. 129-131 of the '840 appendix. Correspondingly, such instructions are identified at pp. 258-260 of the '599 appendix. As the '840 appendix and '599 appendix describe in detail, each of these instructions multiplies a first plurality of floating-point operands found in a first register (e.g., register "ra") by a second plurality of floating-point operands found in a second register (e.g., register "rb") to produce a plurality of products, which

[4] The prior Office Action dated 2/21/07 presented rejections based on references such as Cohen (U.S. Patent No. 5,751,614) and Deip (article entitled "Performance Evaluation of the PowerPC 620 Microarchitecture"). In the response filed 11/15/06, Applicants set forth evidence to show that Cohen and Deip cannot qualify as prior art, by providing a detailed explanation of support for the claims as found in the '840 and '599 disclosures, to establish the priority date of 8/16/95 for the present invention. In the current Office Action dated 10/30/07, the rejections based on Cohen and Deip have been withdrawn.

[5] See Office Action dated 10/30/07, p. 11, "Response to Arguments" section.

are provided to partitioned fields of a result register as a catenated result (e.g., register "rc"). See '840 appendix, p. 130. Also see '599 appendix, p. 260.

Claims 24 and 31 each recites limitations relating to "wherein each of the first and second operands has a width of 64 bits." The '840 and '599 disclosures both describe examples of such instructions where each of the first and second operands has a width of 64 bits. Such an instruction is the "GF.MUL.64" (Group floating-point multiply double) instruction. See '840 appendix at p. 129-131. Also see '599 appendix at pp. 258-260.

Claims 25 and 32 each recites limitations relating to "executing a plurality of different group floating-point arithmetic operations that arithmetically operate on multiple floating-point operands stored in partitioned fields of registers in the register file to produce a catenated result that is returned to a register in the register file, wherein the catenated result comprises a plurality of individual floating-point results." The '840 and '599 disclosures clearly describe embodiments of such instructions. Both the '840 appendix and the '599 appendix describe embodiments of group floating-point arithmetic operations that arithmetically operate on multiple floating-point operands stored in partitioned fields of registers to produce a catenated result that is returned to a register, as recited in claims 25 and 32. At least three such instructions are described: (1) "GF.MUL.16" (Group floating-point multiply half), (2) "GF.MUL.32" (Group floating-point multiply single), and (3) "GF.MUL.64" (Group floating-point multiply double). These instructions are identified at pp. 129-131 of the '840 appendix. Correspondingly, such instructions are identified at pp. 258-260 of the '599 appendix. As the '840 appendix describes in detail, each of these instructions performs group multiplication operations, which are group arithmetic operations, on a first plurality of floating-point operands stored in partitioned fields of a first register (e.g., register "ra") and a second plurality of floating-point operands stored in partitioned fields of a second register (e.g., register "rb") to produce a plurality of floating-point results returned as a catenated result to a register (e.g., register "rc"). See '840 appendix, p. 130. Also see '599 appendix, p. 260.

Support for these claims as found in the '840 and '599 disclosures represent detailed descriptions of specific embodiments that satisfy both the written description and enablement requirements of 35 U.S.C. § 112, paragraph 1. The support found in the '840 and '599 disclosures fully describes the operations of the various instructions, down to the bit level. One of ordinary skill in the art would reasonably conclude, upon reviewing these disclosures, that the inventor had possession of the claimed invention. These disclosures would also enable one of ordinary skill in the art to make and use the claimed invention. This is because the '840 and '599 disclosures define each of the identified instructions in a manner that specifies the precise operation of the instruction, including exactly how every bit of data is obtained (e.g., from specific registers), operated on, and presented as output. One of ordinary skill in the art would not be required to perform undue experimentation – or any experimentation at all for that matter – to understand exactly how the "GF.MUL.16," "GF.MUL.32," and "GF.MUL.64" instructions are carried out as illustrative embodiments of the invention.

In addition, Mr. Korbin Van Dyke's declaration is submitted as part of this Response as further evidence that the '840 and '599 disclosures provide support for claims 9, 18, 24, 25, 31, and 32. In his declaration, Mr. Van Dyke presents a thorough, step-by-step analysis of the support found in the '840 and '599 disclosures, which leads to his conclusion that these disclosures satisfy both the written disclosure and the enablement requirements. Thus, Mr. Van Dyke states in his declaration at p. 18, paragraphs 53-54, that:

> 53. The '840 patent, including the Terpsichore manual, provides sufficient information in sufficient detail describing the claimed invention (as amended) of the 10/757,516 patent application, that one of ordinary skill in the art would reasonably conclude that the inventors had possession of the claimed invention at the time of filing the '840 patent. Further, the '840 patent, including the Terpsichore manual, provides sufficient information regarding the subject matter of the claimed invention (as amended) of the 10/757,516 patent application to enable one of ordinary skill in the pertinent art to make and use the claimed invention without undue experimentation. In addition, the '599 patent, including the Zeus manual, provides sufficient information in sufficient detail describing the claimed invention (as amended) of the 10/757,516 patent application, that one of ordinary skill in the art would reasonably conclude that the inventors had possession of the claimed invention at the time of filing the '599 patent. Further, the '599 patent, including the Zeus manual, provides sufficient information regarding the subject matter of the claimed invention (as amended) of the 10/757,516 patent application to enable one of ordinary skill in the pertinent art to make and use the claimed invention without undue experimentation.
>
> 54. Therefore, I believe that each of the '840 patent, including the Terpsichore manual, and the '599 patent, including the Zeus manual, provide adequate written

> description and enablement as required by 35 USC § 112 for the limitations of claims 9, 18, 24, 25, 31, and 32 (as amended) of the 10/757,516 patent application, as discussed in paragraph 3 of this declaration.

The previous discussions and the detailed declaration from Mr. Korbin Van Dyke clearly demonstrate that the '840 and '599 disclosures indeed provide proper support for claims 9, 18, 24, 25, 31, and 32, to establish the 8/16/95 priority date of the present application. The earliest possible effective filing date of the cited reference Kabir occurs later, on 11/27/95. As such, Kabir cannot qualify as prior art against the present invention. For at least this additional reason, the rejection of claims 9, 18, 24, 25, 31, and 32 should be withdrawn.

CONCLUSION

Applicants respectfully submit that the rejection of claims 1-32 should be withdrawn. As amended, the claims recite features that clearly distinguish over the cited reference, Hao. Hao fails to disclose and in fact teaches away from these claimed features. Furthermore, the rejection of 9, 18, 24, 25, 31, and 32 should be withdrawn for the additional reason that the secondary reference cited in the rejection of these claims, Kabir, cannot qualify as prior art.

In view of the foregoing, Applicants believe all claims now pending in this Application are in condition for allowance. The issuance of a formal Notice of Allowance at an early date is respectfully requested. If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at 202-756-8000.

To the extent necessary, a petition for an extension of time under 37 C.F.R. 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account 500417 and please credit any excess fees to such deposit account.

Respectfully submitted,

McDERMOTT WILL & EMERY LLP

Demetria A. Buncum
Registration No. 58,848

600 13th Street, N.W.
Washington, DC 20005-3096
Phone: 202.756.8000 DAB:amz
Facsimile: 202.756.8087
Date: April 30, 2008

Please recognize our Customer No. 20277 as our correspondence address.